PAPER

A Tight Upper Bound on Online Buffer Management for Multi-Queue Switches with Bicodal Buffers

Koji KOBAYASHI[†a)], ***Nonmember***, **Shuichi MIYAZAKI**[††b)], ***and*** **Yasuo OKABE**[††c)], ***Members***

SUMMARY The online buffer management problem formulates the problem of queuing policies of network switches supporting QoS (Quality of Service) guarantee. In this paper, we consider one of the most standard models, called multi-queue switches model. In this model, Albers et al. gave a lower bound $\frac{e}{e-1}$, and Azar et al. gave an upper bound $\frac{e}{e-1}$ on the competitive ratio when m, the number of input ports, is large. They are tight, but there still remains a gap for small m. In this paper, we consider the case where $m = 2$, namely, a switch is equipped with two ports, which is called a bicordal buffer model. We propose an online algorithm called Segmental Greedy Algorithm (SG) and show that its competitive ratio is at most $\frac{16}{13}(\simeq 1.231)$, improving the previous upper bound by $\frac{9}{7}(\simeq 1.286)$. This matches the lower bound given by Schmidt.

key words: *competitive analysis, multi-queue switches, buffer management*

1. Introduction

When we consider the performance of Internet traffic, one of the crucial problems is a buffer management for routers or switches. The task of a switch is to receive a packet, find its destination, and transmit it from an appropriate output port. However, when the arrival rate of packets exceeds the transmission capacity of a switch, some packets will be lost. To ease this situation, buffers are introduced; when an arrival rate of packets is bursty, we temporary store those packets to buffers and process them when available. One of the key strategies in managing buffers is to decide the acceptance of packets. For example, we are to decide whether to accept the current packet, or to reject it for more important ones that may arrive in the future.

Recently, this kind of problem is formulated as online problems, and a great amount of work has been done. Many models have been proposed, and the most basic one is the following [1]: A switch is equipped with a buffer of bounded size B. An input is a sequence of events. Each event is an arrival event or a send event. At an arrival event, one packet arrives at an input port. Each packet is of unit size and has a value that represents its priority. A buffer can store packets provided that the total size of stored packets does not exceed B, namely, a switch can store up to B packets at the same time. At an arrival event, if the buffer is full, the new packet is rejected. If there is a room for the new packet, an online policy determines, without knowledge of the future, to accept it or not. At each send event, the packet at the head of the queue is transmitted. The goal of the problem is to maximize the sum of the values of transmitted packets. The goodness of an online algorithm is evaluated by the competitive analysis [8], [12]. If, for any input σ, an online algorithm ALG gains value at least $1/c$ of the optimal offline policy for σ, then we say that ALG is c-competitive.

Up to the present, several models have been considered. Among them, Azar et al. have introduced the Multi-Queue Switches model [5]. In this model, a switch consists of m input ports and one output port, and each packet has a destination port. Each port has a buffer (FIFO queue), which can simultaneously store up to B packets. An input is a sequence of events. Each event is an arrival event or a scheduling event (which is similar to the send event described above). When a packet arrives at an arrival event, an online algorithm determines to accept it (if the buffer has room for the new packet) or reject it. The value of an arriving packet is unit. Hence, there is no need to preempt a packet since all packets have the same size and value. At a scheduling event, an algorithm selects one nonempty buffer and transmits the packet at the head of the queue through the output port.

Previous Results and Our Results. In the Multi-Queue Switches model, Albers et al. [3] gave a lower bound of $e/(e-1)(\simeq 1.581)$ for deteministic online algorithms for any B and large enough m. On the other hand, Azar et al. [4] showed an $e/(e-1)$-competitive deterministic algorithm for $B > \log m$. Hence, the upper and lower bounds match for large m, but there still remains a gap for small m. Usually, the performance of an algorithm is evaluated by its asymptotic behavior, e.g., when m goes infinity. However, it is natural to assume that the number of output ports m is bounded in real-world network, and hence, it is important to improve the competitive ratio in the case that m is constant.

For the case of $m = 2$, which is called *bicodal buffers*, Schmidt [11] presented a $9/7(\simeq 1.286)$-competitive deterministic algorithm and proved that the competitive ratio of any deterministic algorithm is at least $16/13(\simeq 1.230)$ for large enough B. Very recently, Bienkowski [7] presented a $16/13$-competitive randomized algorithm, but the deterministic case remains an resolved issue.

In this paper, we improve the upper bound from $9/7$ to $16/13$, which matches the lower bound given by Schmidt [11]. Let us briefly explain an idea of improvement.

Manuscript received June 13, 2008.

Manuscript revised August 11, 2008.

[†]The author is with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.

[††]The authors are with the Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606–8501 Japan.

a) E-mail: kobaya@net.ist.i.kyoto-u.ac.jp

b) E-mail: shuichi@media.kyoto-u.ac.jp

c) E-mail: okabe@media.kyoto-u.ac.jp

DOI: 10.1093/ietisy/e91–d.12.2757

Our algorithm Segmental Greedy Algorithm (SG) divides each queue into two segments S_0 and S_1, and we estimate the number of packets transmitted from S_0 and S_1 independently. For convenience, we also divide queues of OPT in the same way as SG and compare the number of transmitted packets. We first fix an input σ, and show that there exists a desirable optimal offline algorithm OPT^* (depending on σ) that transmits the same number of packets from S_0 with SG. For the analysis of S_1 part, we modify σ and construct another input σ'. Simply speaking, σ' is constructed from σ by using only packets processed by SG in S_1. To link the number of packets transmitted from S_1 by OPT^* and SG, we analyze the performance of greedy algorithm for σ'.

Related Results. Several results on the competitiveness of the unit-value multi-queue model have been presented [3]–[5]. Azar et al. [5] gave a lower bound of $1.366 - \Theta(1/m)$ for a deterministic algorithm for any B. Albers et al. [3] showed that the competitive ratio of any greedy algorithms is at least $2-1/B-\Theta(m^{-1/(2B-2)})$ for any B and large enough m. They also gave a $17/9(\simeq 1.89)$-competitive deterministic algorithm for $B \geq 2$, and it is optimal in the case $B = 2$. Albers and Jacobs [2] performed an experimental study on several online algorithms for the multi queue model for the first time.

Much work has also been done for the case of multi-value multi-queue models. In this model, $\alpha(\geq 1)$ is the ratio between the largest and the smallest values of packets. Azar et al. [5] gave a lower bound of $1.366 - \Theta(1/m)$ for a deterministic algorithm for any B. Azar et al. [6] gave a 3-competitive deterministic algorithm for the preemptive case. Itoh et al. [9] showed no non-preemptive algorithm can be better than $1 + 1/(\alpha \ln(\alpha/(\alpha - 1)))$-competitive. For the 2-value multi-queue model, Itoh et al. [10] presented an upper bound $3 - 1/\alpha$ for the preemptive case. They also showed that the competitive ratio of any online algorithms is at least $1.514 - \Theta(0.559^m)$.

2. Preliminaries

In this section, we formally define the problem studied in this paper, which was originally proposed in [5]. Then, we present Segmental Greedy Algorithm(SG) for this model with 2 input ports.

2.1 Online Buffer Management Problem for Multi-Queue Switches

A multi-queue switch has m input ports (FIFO queues) each of which is equipped with a buffer whose size is B. The size of a packet is one, hence each port can store up to B packets simultaneously. All m queues are empty at the beginning. The value of any packet is unit. In this paper, we consider a switch with 2 input ports, namely $m = 2$, called *bicodal buffers*.

An input is a sequence of events. An *event* is an *arrival event* or a *scheduling event*. At an arrival event, a packet (say, p) arrives at an input port (1 through m), and the task of an online algorithm (or an online policy) is to select one of the following actions: insert an arriving packet into the corresponding queue (*accept p*), or drop it (*reject p*). If a packet is accepted, it is stored at the tail of the corresponding input queue. Since the value of all packets are the same, we may assume that an arriving packet is accepted greedily if the corresponding queue has a space. Further, we assume that no more than one packets arrive at the same time. At a scheduling event, an online algorithm selects one nonempty input port from m ones and transmits the packet at the head of the queue.

The *gain* of an algorithm is the sum of the number of transmitted packets. Therefore, our goal is to maximize the sum of the values of packets eventually transmitted. The cost of an algorithm ALG for an input σ is denoted by $T_{ALG}(\sigma)$. If $T_{ALG}(\sigma) \geq T_{OPT}(\sigma)/c$ for an arbitrary input σ, we say that ALG is *c-competitive*, where OPT is an optimal offline policy for σ. Also, we do not consider inputs including packets which both OPT and an online algorithm reject. For simplicity of analysis, we consider the algorithm which transmits a packet at a scheduling event whenever its buffer is not empty. Such an algorithm is called *work-conserving*. (See [5], e.g.) Also, we assume that no arrival event happens once both of an online algorithm's buffers become empty. These assumptions do not affect the analysis of the competitive ratio.

For analysis, we give following definitions about buffers. Since a value of each packet is unit, we do not need to distinguish packets in buffers. Hence, we assume that an algorithm can transmit an arbitrary packet in the buffer at a scheduling event. In addition, we assign index numbers 1 through B to each position of a buffer from the head in an increasing order. Also, the ith queue of the switch is denoted as $Q^{(i)} (1 \leq i \leq m)$, and the jth position of $Q^{(i)}$ is denoted as $Q^{(i,j)} (1 \leq j \leq B)$, which is called the jth *cell* of $Q^{(i)}$.

2.2 Segmental Greedy Algorithm (SG)

In this section, we give the definition of Segmental Greedy Algorithm (SG), which we propose in this paper.

We give some definitions. For time t when an event occurs, $t-$ represents the moment before t and after the previous event occurred. Similarly, $t+$ is the moment after t and before the next event occurs. For an algorithm A, $\ell_A^{(i,j)}(t)$ is a boolean variable that is 1 if A holds a packet in $Q^{(i,j)}$ at time t when no event happens, and 0 otherwise. Even if $\ell_A^{(i,j)}(t)$ is a boolean variable, for convenience, we may sometimes use it as natural numbers. x, k_1, k_2, k_3 and τ are internal variables which SG uses. SG's execution is determined by k_1, k_2 and k_3. Actually, k_1, k_2, and k_3 are redundant for specifying the execution of SG; we may define $k = \sum_{j=1}^{3} k_j$ and use it instead of k_1, k_2, and k_3. However, in the later analysis, we need to count the number of executions of some cases of SG. For this purpose, we use these three variables separately. Intuitively, $\sum_{j=1}^{3} k_j$ means the number of packets which OPT can transmit from S_1. τ is initially 0, and increases only at an arrival event. This is a monotonically non-decreasing value that is used in SG as follows: For time t when an event does not happen, let $\mathcal{T}(t)$ denote the value of τ at time t. Using $\mathcal{T}(t)$, we divide each queue $Q^{(i)}$ ($i = 1, 2$) of SG into two segments as follows (note that the way of division changes according to time): Positions of $Q^{(i)}$ which have index numbers $\mathcal{T}(t) + 1$ through B (1 through $\mathcal{T}(t)$, respectively) is called a *Segment 0*(S_0 for short) (Segment 1 (S_1 for short), respectively) at time t. $H^{(i)}_{SG,j}(t)$ denotes the number of packets SG holds in $Q^{(i)}$ on S_j at time t when no event happens, namely, $H^{(i)}_{SG,0}(t) = \sum_{j=\mathcal{T}(t)+1}^{B} \ell^{(i,j)}_{SG}(t)$ and $H^{(i)}_{SG,1}(t) = \sum_{j=1}^{\mathcal{T}(t)} \ell^{(i,j)}_{SG}(t)$.

Without loss of generality, we assume that when SG transmits a packet from S_0 (S_1, respectively) of $Q^{(i)}$, it transmits a packet at the position with the smallest (largest, respectively) index. We now show the definition of SG.

An execution of SG is simple. At an arrival event, the arriving packet is accepted to S_0 if there exists an empty cell in S_0. Otherwise, it is accepted into S_1 if there exists an empty cell in S_1. Otherwise, it is rejected. At a scheduling event, a packet is greedily transmitted from S_1 if $\sum_{j=1}^{3} k_j > 0$, where "greedy" means to send a packet from the queue having the more packets (see the description of the algorithm for the precise definition). Otherwise, a packet is greedily transmitted from S_0 if there exists a packet in S_0. Otherwise, a packet is greedily transmitted from S_1. For better understanding, we put an example of the execution of SG in Appendix A.

Segmental Greedy Algorithm (SG)
Initialize: $\tau := 0$, $k_1 := k_2 := k_3 := 0$.

Arrival event at time t (Let p be a packet arriving at $Q^{(i)}$.)
Step A1: If $\tau \neq B$, do the following:

> Checking S_0 Step:
> Initialize: $x := \tau + 1$
> Case A1.1 ($\ell^{(i,x)}_{SG}(t-) = 0$):
> Accept p to $Q^{(i,x)}$,
> and execute one of the following cases.
> Case A1.1.1 ($\ell^{(j,x)}_{SG}(t-) = 1$ ($j \neq i$)):
> $k_3 := k_3 + 1$, $\tau := \tau + 1$, and stop.
> (We call a packet at $Q^{(j,x)}$ *mate* of p.)
> Case A1.1.2 ($\ell^{(j,x)}_{SG}(t-) = 0$ ($j \neq i$)): Stop.
> Case A1.2 ($\ell^{(i,x)}_{SG}(t-) = 1$):
> Execute one of the following cases.
> Case A1.2.1 ($x = B$ and $\tau \neq 0$):
> Go to Step A2.
> Case A1.2.2 ($x = B$ and $\tau = 0$):
> Reject p, and stop.
> Case A1.2.3 ($x < B$):
> $x := x + 1$ and go to Case A1.1.

Step A2: Do the following:

> Checking S_1 Step:
> Initialize: $x := \tau$
> Case A2.1 ($\ell^{(i,x)}_{SG}(t-) = 0$):
> Accept p to $Q^{(i,x)}$, $k_2 := k_2 + 1$, and stop.
> Case A2.2 ($\ell^{(i,x)}_{SG}(t-) = 1$):
> Execute one of the following cases.
> Case A2.2.1 ($x = 1$):
> Reject p, $k_1 := \tau, k_2 := 0, k_3 := 0$, and stop.
> Case A2.2.2 ($x > 1$):
> $x := x - 1$ and go to Case A2.1.

Scheduling event at time t
Case S1.1 ($k_1 + k_2 + k_3 > 0$):
(Note $\sum_{j=1}^{2} H^{(j)}_{SG,1} > 0$ in this case.
See Lemma Appendix B.2.)
Execute Greedy Step (see below),
and then execute one of the following cases.
Case S1.1.1 ($k_1 > 0$):
$k_1 := k_1 - 1$ and stop.
Case S1.1.2 ($k_1 = 0$ and $k_2 > 0$):
$k_2 := k_2 - 1$ and stop.
Case S1.1.3 ($k_1 = k_2 = 0$ and $k_3 > 0$):
$k_3 := k_3 - 1$ and stop.
Case S1.2 ($k_1 = k_2 = k_3 = 0$):
Execute one of the following cases.
Case S1.2.1 ($H^{(1)}_{SG,0}(t-) > 0$ or $H^{(2)}_{SG,0}(t-) > 0$):
(Note that only one queue can have a packet in S_0 at any time. See Lemma Appendix B.1.)
Select a packet p from S_0 of the non-empty queue, transmit p, and stop.
Case S1.2.2 (Otherwise):
Execute Greedy Step (if possible), and stop.

> Greedy Step:
> If $H^{(1)}_{SG,1}(t-) \geq H^{(2)}_{SG,1}(t-)$, transmit a packet from S_1 of $Q^{(1)}$. Otherwise, namely, if $H^{(1)}_{SG,1}(t-) < H^{(2)}_{SG,1}(t-)$, transmit a packet from S_1 of $Q^{(2)}$.

Here we give one remark on Step A1.2.2. Step A1.2.2 is executed for general inputs. However, we later restrict inputs in Sec. 3.2 to simplify analysis. For such inputs, Step A1.2.2 is never executed. The reason is as follows: If $\tau = 0$ when Step A1.2.2 is executed at time t, namely, a whole buffer is S_0 in both queues at t, SG always executes Step A1.1.2 at each arrival event between 0 and t by the definition of SG. Therefore, if there exists a packet at $Q^{(a)}$ ($a = 1, 2$) between 0 and t, there does not exist a packet at $Q^{(b)}$ ($a \neq b$). Hence, at a scheduling event between 0 and t, OPT and SG always transmit a packet from the same queue. Therfore, since the buffer of OPT is full if that of SG is full, OPT rejects an arriving packet when SG executes A1.2.2 and reject it.

3. 16/13 Upper Bound

3.1 Overview of the Analysis

Recall that SG divides each queue into two segments. For the purpose of analysis, we divide each queue $Q^{(i)}$ ($i = 1, 2$) of OPT, in the same way as we have done for SG, namely, positions of $Q^{(i)}$ which have index numbers $\mathcal{T}(t)+1$ through B (1 through $\mathcal{T}(t)$, respectively) is called a *Segment 0* (S_0) (Segment 1 (S_1), respectively) at time t. Note that $\mathcal{T}(t)$ here is the variable used by SG. Hence, at any time, the size of each segment is the same in SG and OPT. For an input σ and an algorithm A, $F_{A,j}(\sigma)$ ($j = 0, 1$) denotes the total number of packets which A transmits from S_j. Using $F_{A,i}(\sigma)$, we can write $T_A(\sigma) = F_{A,0}(\sigma) + F_{A,1}(\sigma)$.

For analysis, we first fix an arbitrary input σ. In Sec. 3.2, we show that there exists a desirable optimal offline algorithm OPT^* for σ that satisfies $F_{OPT^*,0}(\sigma) = F_{SG,0}(\sigma)$. In Sec. 3.3, we prove that $F_{OPT^*,1}(\sigma) \leq \frac{16}{13}F_{SG,1}(\sigma)$. Therefore, $T_{OPT^*}(\sigma) = F_{OPT^*,0}(\sigma) + F_{OPT^*,1}(\sigma) \leq F_{SG,0}(\sigma) + \frac{16}{13}F_{SG,1}(\sigma) \leq \frac{16}{13}(F_{SG,0}(\sigma) + F_{SG,1}(\sigma)) = \frac{16}{13}T_{SG}(\sigma)$. Hence, we have the following theorem:

Theorem 3.1: The competitive ratio of SG is at most 16/13.

3.2 Evaluating S_0

At first, we restrict the input for simplicity of analysis. For an algorithm A, $h_A^{(i)}(t)$ denotes the number of packets $Q^{(i)}$ holds at time t when no event happens.

Lemma 3.2: Let ON be an online algorithm. For any input σ, there exists another input σ' that satisfies the following (i) and (ii): (i) $\frac{T_{OPT}(\sigma)}{T_{ON}(\sigma)} \leq \frac{T_{OPT}(\sigma')}{T_{ON}(\sigma')}$. (ii) Let t be an arbitrary time when arriving event happens at which ON rejects a packet, and suppose that this packet is destined for $Q^{(i)}$. Also, let t' be the scheduling time immediately after t. Then, $h_{ON}^{(i)}(t'-) = h_{OPT}^{(i)}(t'-) = B$.

Proof. We assume that $h_{ON}^{(j)}(t'-) - h_{OPT}^{(j)}(t'-) = x$, and construct σ' from σ as follows: (i) σ' includes all events in σ. (ii) Add x arrival events in (t, t') at which a packet arrives at $Q^{(i)}$. (iii) Note that OPT may reject arriving packets after t' because of the operation (ii) above, and this happens at most x times. Remove all such arrival events from σ', and let y be the number of these removed arrival events. (iv) Add $x - y$ scheduling events to the end of σ'. Then, $h_{ON}^{(i)}(t'-) = h_{OPT}^{(i)}(t'-) = B$ holds. Also, since OPT accepts $x - y$ new arriving packets among x packets added by (ii) for σ', and can transmit them at scheduling events by (iv), $T_{OPT}(\sigma') = T_{OPT}(\sigma) + x - y$. On the other hand, ON can accept none of x new arriving packets by (ii), and if OPT reject a packet p at an arrival event, ON cannot accept p. Hence, $T_{ON}(\sigma') = T_{ON}(\sigma)$. Therefore, $\frac{T_{OPT}(\sigma)}{T_{ON}(\sigma)} \leq \frac{T_{OPT}(\sigma')}{T_{ON}(\sigma')}$. □

By Lemma 3.2, we may consider only inputs that satisfy the following: For any arrival event when an online algorithm ON rejects a packet destined for $Q^{(i)}$, $h_{ON}^{(i)}(t'-) = h_{OPT}^{(i)}(t'-) = B$ holds, where $t'(> t)$ is the time for the scheduling event that happens immediately after t.

Without loss of generality, for the purpose of analysis, we assume that OPT behaves in the same way as SG at arrival events, namely, OPT accepts an arriving packet p into S_0 if S_0 has a room. Otherwise, namely, if S_0 is full, OPT accepts p into S_1. Also, we assume that when OPT accepts a packet into S_0 (S_1, respectively) of $Q^{(i)}$, it stores a packet at the position with the smallest (largest, respectively) index. Furthermore, when OPT transmits a packet from S_0 (S_1, respectively) of $Q^{(i)}$, it transmits a packet at the position with the largest (smallest, respectively) index.

We impose the following rule to OPT's behavior at scheduling events.

The Synchronizing Rule: Let t be a time for a scheduling event when both SG and OPT transmit a packet, and suppose that OPT transmits a packet from $Q^{(i)}$ at t. Then, OPT decides the segment from which a packet is transmitted according to the behavior of SG at t. (Note that this rule is only for the purpose of analysis, and does not affect the performance of OPT.) Namely, if SG transmits from S_j at t, then OPT also transmits a packet from S_j if OPT has at least one packet in S_j of $Q^{(i)}$. (In this case, we say that OPT and SG *synchronize* at t.) If OPT does not have a packet in S_j of $Q^{(i)}$, OPT transmits a packet from S_{1-j} of $Q^{(i)}$. We say that OPT and SG synchronize within (t', t) if OPT and SG synchronize at all scheduling events that happen within (t', t).

In the following analysis, we prove in Lemma 3.10 that there exists a desirable OPT that can synchronize with SG at any time. Using this lemma, we prove Lemma 3.11 to evaluate the number of packets transmitted from S_0 by OPT and SG. For this purpose, in the following lemmas, we show some properties that hold within a period when OPT and SG synchronize.

Lemma 3.3: Let t' be a time when OPT and SG synchronize within $(0, t')$.
Then, $\forall t < t', \forall i\ H_{OPT,0}^{(i)}(t) = H_{SG,0}^{(i)}(t)$.

Proof. We prove the lemma inductively on time. At the beginning, the statement is true since $H_{OPT,0}^{(i)}(0) = H_{SG,0}^{(i)}(0) = 0$. Let $t(< t')$ be a time when an event happens. We assume that the statement is true at time $t-$ and show that it is true at $t+$, namely, we assume that $H_{OPT,0}^{(i)}(t-) = H_{SG,0}^{(i)}(t-)$ and show $H_{OPT,0}^{(i)}(t+) = H_{SG,0}^{(i)}(t+)$.

Case 1. Scheduling event (Case S1.1, S1.2.2). Since SG transmits a packet from S_1 and OPT and SG synchronize at time t, OPT transmits a packet from S_1 also. Therefore, $H_{OPT,0}^{(i)}(t+) = H_{SG,0}^{(i)}(t+)$.

Case 2. Scheduling event (Case S1.2.1). We assume that SG transmits from $Q^{(i)}$. Since SG transmits a packet from S_0, $H_{SG,0}^{(i)}(t+) = H_{SG,0}^{(i)}(t-) - 1$. Since $H_{SG,0}^{(i)}(t-) > 0$ and at least one S_0 of SG is empty, $H_{SG,0}^{(j)}(t-) = 0$ ($i \neq j$).

So, by the induction hypothesis, $H^{(j)}_{OPT,0}(t-) = 0$. Since OPT transmits a packet from S_0 according to the Synchronizing Rule, it must send from $Q^{(i)}$ (since S_0 of $Q^{(i)}$ is empty), and hence $H^{(i)}_{OPT,0}(t+) = H^{(i)}_{OPT,0}(t-) - 1$. Therefore, $H^{(i)}_{OPT,0}(t+) = H^{(i)}_{SG,0}(t+)$ and $H^{(j)}_{OPT,0}(t+) = H^{(j)}_{SG,0}(t+)$.

We then consider the case that the event at t is an arrival event. Assume that an arriving packet p is destined for $Q^{(i)}$. We consider the following three cases.

Case 3. Arrival event (Case A1.1.1). Since $\ell^{(j,\mathcal{T}(t+))}_{SG}(t-) = 1$ $(j \neq i)$, $H^{(j)}_{SG,0}(t-) > 0$ and $H^{(i)}_{SG,0}(t-) = 0$ by the definition of SG. Also, since $\mathcal{T}(t+) = \mathcal{T}(t-) + 1$ and SG accepts p into $Q^{(i,\mathcal{T}(t+))}$, $H^{(j)}_{SG,0}(t+) = H^{(j)}_{SG,0}(t-) - 1$ and $H^{(i)}_{SG,0}(t+) = H^{(i)}_{SG,0}(t-)$ hold. By the induction hypothesis, $H^{(j)}_{OPT,0}(t-) = H^{(j)}_{SG,0}(t-)$ and $H^{(i)}_{OPT,0}(t-) = H^{(i)}_{SG,0}(t-)$. Since OPT decides the position into which p is stored in the same way as SG, OPT accepts p into $Q^{(i,\mathcal{T}(t+))}$. Hence, $H^{(j)}_{OPT,0}(t+) = H^{(j)}_{OPT,0}(t-) - 1$ and $H^{(i)}_{OPT,0}(t+) = H^{(i)}_{OPT,0}(t-)$.

Case 4. Arrival event (Case A1.1.2). Since $\ell^{(j,\mathcal{T}(t+))}_{SG}(t-) = 0$ by the definition of SG, $H^{(j)}_{SG,0}(t-) = 0$ and $H^{(i)}_{SG,0}(t-) \geq 0$. Also, $H^{(i)}_{SG,0}(t+) = H^{(i)}_{SG,0}(t-) + 1$ and $H^{(j)}_{SG,0}(t+) = H^{(j)}_{SG,0}(t-)$ hold. $H^{(j)}_{OPT,0}(t-) = H^{(j)}_{SG,0}(t-)$ and $H^{(i)}_{OPT,0}(t-) = H^{(i)}_{SG,0}(t-)$ by the induction hypothesis. Hence, since OPT accepts p into S_0 of $Q^{(i)}$ similarly to SG, $H^{(i)}_{OPT,0}(t+) = H^{(i)}_{OPT,0}(t-) + 1$ and $H^{(j)}_{OPT,0}(t+) = H^{(j)}_{OPT,0}(t-)$. Therefore, $H^{(j)}_{OPT,0}(t+) = H^{(j)}_{SG,0}(t+)$ and $H^{(i)}_{OPT,0}(t+) = H^{(i)}_{SG,0}(t+)$.

Case 5. Arrival event (Case A2.1, A2.2). By the induction hypothesis, $H^{(i)}_{OPT,0}(t-) = H^{(i)}_{SG,0}(t-) = B - \mathcal{T}(t-)$ and $H^{(j)}_{OPT,0}(t-) = H^{(j)}_{SG,0}(t-) = 0$. Since SG accepts p into S_1 of $Q^{(i)}$, the number of packets in S_0 of SG does not change. Since S_0 of $Q^{(i)}$ of OPT is also full, OPT accepts p into S_1 and the number of packets in S_0 does not change. Therefore, $H^{(i)}_{OPT,0}(t+) = H^{(i)}_{SG,0}(t+) = B - \mathcal{T}(t-)$ and $H^{(j)}_{OPT,0}(t+) = H^{(j)}_{SG,0}(t+) = 0$.

We have shown that the statement is true at time $t+$. □

For each t when no event happens, we denote the value of k_j $(j = 1, 2, 3)$ at t by $\mathcal{K}_j(t)$.

Lemma 3.4: Let t and t' $(t' < t)$ be times, and $b \in \{0, 1\}$ such that (i) OPT and SG synchronize within $(0, t)$, (ii) $\sum_{j=1}^{3} \mathcal{K}_j(t) = 0$, and (iii) OPT does not transmit a packet from S_1 of $Q^{(b)}$ during (t', t). Then, if a packet p arrives at $t'' \in (t', t)$ such that $H^{(a)}_{OPT,1}(t''+) = H^{(a)}_{OPT,1}(t''-) + 1$ $(a \neq b)$, OPT transmits p within (t'', t).

Proof. Let a-packet be a packet which arrives at $Q^{(a)}$ within (t', t) and is accepted to S_1. Let t-packet be a packet transmitted from S_1 within (t', t). Let x, y and y' be the numbers of a-packets, t-packets which are a-packets, and t-packets which are not a-packets, respectively. Obviously, $y \leq x$. By the condition (iii), every t-packet is transmitted from S_1 of $Q^{(a)}$. When an a-packet arrives at $\tilde{t} \in (t', t)$, $\sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}+) \geq \sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}-) + 1$ holds because SG executes one of Cases A1.1.1, A2.1, and A2.2 at $\tilde{t}$. By the condition (ii) and the above inequality, since SG transmits packets in LIFO order, $\max\{\sum_{j=1}^{3} \mathcal{K}_j(t'+) - y', 0\} + \max\{x - y, 0\} \leq \sum_{j=1}^{3} \mathcal{K}_j(t) = 0$. Therefore, $y \geq x$, and hence $x = y$, which means that all a-packets are also t-packets. This completes the proof. □

Using Lemma 3.4, we have the following corollary.

Corollary 3.5: Let t be a time, and $b \in \{0, 1\}$ such that (i) OPT and SG synchronize within $(0, t)$, (ii) $\sum_{j=1}^{3} \mathcal{K}_j(t) = 0$, and (iii) OPT does not transmit a packet from S_1 of $Q^{(b)}$ during $(0, t)$. Then, $H^{(a)}_{OPT,1}(t-) = 0$ $(a \neq b)$.

Proof. By Lemma 3.4, if a packet p arrives at $Q^{(a)}$ at time $t'(< t)$, OPT transmits p within (t', t). Therefore, $H^{(a)}_{OPT,1}(0) = H^{(a)}_{OPT,1}(t-) = 0$. □

Next, we show a relation between $2\mathcal{T}(t) - \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t)$ and $\mathcal{T}(t) - \sum_{j=1}^{3} \mathcal{K}_j(t)$ when OPT and SG synchronize within $(0, t)$. The former is the number of empty cells of S_1 of OPT's buffer at time t, and the latter is the lower bound on the number of packets transmitted from S_1 before t.

Lemma 3.6: Let t' be a time when OPT and SG synchronize within $(0, t')$. Then, $\forall t < t'$ $\mathcal{T}(t) + \sum_{j=1}^{3} \mathcal{K}_j(t) \geq \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t)$.

Proof. We prove the lemma by induction on time. At the beginning, the statement is true since $\mathcal{T}(0) + \sum_{j=1}^{3} \mathcal{K}_j(0) = 0$ and $\sum_{i=1}^{2} H^{(i)}_{OPT,1}(t) = 0$. Let $t(< t')$ be a time when an event happens. We assume that the statement is true at time $t-$ and show that it is true at $t+$, namely, we assume that $\mathcal{T}(t-) + \sum_{j=1}^{3} \mathcal{K}_j(t-) \geq \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t-)$ and prove $\mathcal{T}(t+) + \sum_{j=1}^{3} \mathcal{K}_j(t+) \geq \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t+)$. We will consider six cases according to the execution of SG.

We first consider the cases where an arrival event happens.

Case 1. Arrival event (Case A1.1.2). $\sum_{i=1}^{2} H^{(i)}_{OPT,1}(t+) = \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t-)$, $\mathcal{T}(t+) = \mathcal{T}(t-)$, and $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-)$ hold. From the above equalities and the induction hypothesis, $\mathcal{T}(t+) + \sum_{j=1}^{3} \mathcal{K}_j(t+) \geq \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t+)$.

Case 2. Arrival event (Case A1.1.1). Since $\sum_{i=1}^{2} H^{(i)}_{OPT,1}(t+) = \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t-) + 2$, $\mathcal{T}(t+) = \mathcal{T}(t-) + 1$, and $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-) + 1$, by the induction hypothesis, the statement is true.

Case 3. Arrival event (Case A2.1). We have that $\sum_{i=1}^{2} H^{(i)}_{OPT,1}(t+) = \sum_{i=1}^{2} H^{(i)}_{OPT,1}(t-) + 1$, $\mathcal{T}(t+) = \mathcal{T}(t-)$, and $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-) + 1$. By the above equalities and the induction hypothesis, the statement is true.

Case 4. Arrival event (Case A2.2.1). $\forall i$ $H^{(i)}_{OPT,1}(t+) \leq \mathcal{T}(t+)$ and $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \mathcal{T}(t+)$ hold. Hence, the statement is true.

Next, we consider the case where a scheduling event happens at t. Note that $\mathcal{T}(t-) = \mathcal{T}(t+)$ by the definition of SG.

Case 5. Scheduling event (Case S1.1). $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-) - 1$ holds and SG transmits a packet from S_1. OPT transmits a packet from S_1 also, since SG and OPT synchronize at t. Therefore, $\sum_{i=1}^{2} H_{OPT,1}^{(i)}(t+) = \sum_{i=1}^{2} H_{OPT,1}^{(i)}(t-) - 1$. By the above equalities and the induction hypothesis, the statement is true.

Case 6. Scheduling event (Case S1.2). Since $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-)$, and $\sum_{i=1}^{2} H_{OPT,1}^{(i)}(t+) \leq \sum_{i=1}^{2} H_{OPT,1}^{(i)}(t-)$ hold, by the induction hypothesis, the statement is true. We have shown that the statement holds at $t+$. □

Next, we will prove an important property on $H_{OPT,1}^{(i)}(t)$ when OPT and SG synchronize during $(0, t)$.

Lemma 3.7: Let t and t' ($t' < t$) be times, and $b \in \{0, 1\}$ such that (i) OPT and SG synchronize within $(0, t)$, (ii) $\sum_{j=1}^{3} \mathcal{K}_j(t) = 0$, (iii) OPT does not transmit a packet from S_1 of $Q^{(b)}$ during (t', t), and (iv) $\exists \tilde{t} \in (t', t)$ such that $h_{OPT}^{(b)}(\tilde{t}) = B$. Then, $H_{OPT,1}^{(a)}(t) = 0$ $(a \neq b)$ holds.

Proof. By the condition (iv), $H_{OPT,1}^{(b)}(\tilde{t}) = \mathcal{T}(\tilde{t})$ holds. Now, let $z = H_{OPT,1}^{(a)}(\tilde{t})(\leq \mathcal{T}(\tilde{t}))$ $(a \neq b)$. Then $\sum_{i=1}^{2} H_{OPT,1}^{(i)}(\tilde{t}) = \mathcal{T}(\tilde{t}) + z$. So, using Lemma 3.6, $\mathcal{T}(\tilde{t}) + \sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}) \geq \sum_{i=1}^{2} H_{OPT,1}^{(i)}(\tilde{t})$. From the above two equalities, $\sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}) \geq z$. Now, let a'-packet be a packet which arrives at $Q^{(a)}$ within $(\tilde{t}, t)$ and is accepted to S_1. Let t'-packet be a packet transmitted from S_1 within $(\tilde{t}, t)$. Let x, y and y' be the numbers of a'-packets, t'-packets which are a'-packets, and t'-packets which are not a'-packets, respectively. By the condition (iii), each t'-packet is transmitted from S_1 of $Q^{(a)}$. When an a'-packet arrives at $\tilde{t} \in (t', t)$, $\sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}+) \geq \sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}-) + 1$ holds since SG executes one of Cases A1.1.1, A2.1, and A2.2. By the above definitions, $H_{OPT,1}^{(a)}(t) = z + x - y - y'$ holds. By the condition (ii) and the above inequality, since SG transmits packets in LIFO order, $0 = \sum_{j=1}^{3} \mathcal{K}_j(t) \geq \max\{\sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}) - y', 0\} + \max\{x - y, 0\} \geq \max\{z - y', 0\} + \max\{x - y, 0\}$. Hence, $z \leq y'$ and $x \leq y$. By the above inequalities, $H_{OPT,1}^{(a)}(t) = z + x - y - y' \leq 0$. □

Lemma 3.8: Let t and t' ($t' < t$) be times when scheduling events happen, and $b \in \{0, 1\}$ such that (i) OPT transmits a packet from S_1 of $Q^{(b)}$ at t, (ii) OPT transmits a packet from S_1 of $Q^{(a)}$ $(a \neq b)$ at t', (iii) $\forall t'' \in (t', t)$, $h_{OPT}^{(a)}(t'') \leq B - 1$, and (iv) $\forall t'' \in (t', t)$ such that $H_{OPT,1}^{(b)}(t'') \geq 1$. Consider an online algorithm A that acts as follows: (v) A transmits a packet from S_1 of $Q^{(a)}$ at t, (vi) A transmits a packet from S_1 of $Q^{(b)}$ at t', and (vii) at any scheduling event (other than t and t') A selects the same queue and segment as OPT to transmit a packet. (Note that there is no guarantee that A can transmit a packet whenever OPT transmits.) Then, A is an optimal offline algorithm.

Proof. We evaluate the number of packets transmitted by A and OPT at each time.

First, we consider the period $(0, t')$. By the condition (vii), the number of packets transmitted by A is equal to that by OPT, and $\forall i, j$ $H_{OPT,j}^{(i)}(t'-) = H_{A,j}^{(i)}(t'-)$.

Next, we consider the period (t', t). First, we analyze $Q^{(b)}$. By the conditions (i) and (vi), $H_{A,1}^{(b)}(t'+) = H_{OPT,1}^{(b)}(t'+) - 1$. Also, by the condition (iv), OPT does not transmit a packet from S_1 of $Q^{(b)}$ at time $\tilde{t}$ such that $H_{OPT,1}^{(b)}(\tilde{t}) = 1$ $(\tilde{t} \in (t', t))$. Hence, by the condition (vii), $H_{A,1}^{(b)}(\hat{t}) = H_{OPT,1}^{(b)}(\hat{t}) - 1$ $(\hat{t} \in (t', t))$, and therefore, A can accept all packets which arrive at $Q^{(b)}$ within (t', t), and the number of packets which OPT transmits within $(t', t]$ from $Q^{(b)}$ is the same as the number of packets which A transmits within $[t', t)$ from $Q^{(b)}$. So, $\forall j$ $H_{OPT,j}^{(b)}(t+) = H_{A,j}^{(b)}(t+)$.

We then consider $Q^{(a)}$. By the conditions (ii) and (v), $h_A^{(a)}(t'+) = h_{OPT}^{(a)}(t'+) + 1$ holds, and by the condition (iii), a packet does not arrive at $Q^{(a)}$ at time $\tilde{t}$ $(\tilde{t} \in (t', t))$ such that $h_{OPT}^{(a)}(\tilde{t}) = B - 1$. Hence, A can accept all packets which arrive at $Q^{(a)}$ within (t', t). Also, by the condition (vii), $H_{A,0}^{(a)}(\hat{t}) = H_{OPT,0}^{(a)}(\hat{t})$ $(\hat{t} \in (t', t))$ and $H_{A,1}^{(a)}(\hat{t}) = H_{OPT,1}^{(a)}(\hat{t}) + 1$ $(\hat{t} \in (t', t))$. Hence the number of packets which OPT transmits within $[t', t)$ from $Q^{(a)}$ is equal to the number of packets which A transmits within $(t', t]$ from $Q^{(a)}$, and so, $\forall j$ $H_{OPT,j}^{(a)}(t+) = H_{A,j}^{(a)}(t+)$.

Finally, we consider the period after t. By the condition (vii) and the above equalities, namely, $\forall i, j$ $H_{OPT,j}^{(i)}(t+) = H_{A,j}^{(i)}(t+)$, the number of packets transmitted by A is the same as that by OPT. □

Lemma 3.9: Let t be a time when no event happens, and $a \in \{0, 1\}$ such that (i) OPT and SG synchronize within $(0, t)$, (ii) $\sum_{j=1}^{3} \mathcal{K}_j(t) > 0$, and (iii) $H_{OPT,0}^{(a)}(t) > 0$. Then, $H_{OPT,1}^{(a)}(t) > 0$.

Proof. Let t' be a time when an arrival event happens, and assume that $\forall \tilde{t} \in (t', t)$ $\sum_{j=1}^{3} \mathcal{K}_j(\tilde{t}) > 0$. We first show a few important properties that will be used several times in the following arguments. By the definition of t', the segment from which SG transmits a packet during (t', t) is always S_1. Since OPT and SG synchronize within $(0, t)$ by the condition (i), the segment from which OPT transmits a packet within (t', t) is also S_1. Property (1): Assume that SG executes Case A2.1 or Case A2.2.1 in (t', t), and let $\tilde{t} \in (t', t)$ be a time when SG executes Case A2.1 or Case A2.2.1. Then, note that by the condition (iii), at t OPT still holds some packets existed in S_0 at $\tilde{t}$. Hence $H_{OPT,0}^{(a)}(\tilde{t}+) > 0$. Property (2): By the above property (1) and the condition (i), if SG executes Case A2.1 at time $\hat{t} \in (t', t)$, $H_{A,1}^{(a)}(\hat{t}+) = H_{A,1}^{(a)}(\hat{t}-) + 1$ $(A = OPT, SG)$ since a packet arrives at $Q^{(a)}$ at $\hat{t}$. Property (3): Also, if SG executes Case A1.1.1 at time $\hat{t} \in (t', t)$, $H_{A,1}^{(a)}(\hat{t}+) = H_{A,1}^{(a)}(\hat{t}-) + 1$ $(A = OPT, SG)$ since $\mathcal{T}(\hat{t}+) = \mathcal{T}(\hat{t}-) + 1$.

Case 1. $\mathcal{K}_1(t) > 0$. If $\mathcal{K}_1(t) > 0$, SG executes Case A2.2.1 before t by the definition of SG. Hence, let $\hat{t} \in (t', t]$ be the time when SG executes Case A2.2.1 and does not execute Case A2.2.1 within $(\hat{t}, t)$. Then, by the definition of SG, $\mathcal{K}_1(\hat{t}+) = \mathcal{T}(\hat{t}+)$. Since $h^{(a)}_{OPT}(\hat{t}+) = h^{(a)}_{SG}(\hat{t}+) = B$ by Lemma 3.2 and the above property (1), $H^{(a)}_{OPT,1}(\hat{t}+) = H^{(a)}_{SG,1}(\hat{t}+) = \mathcal{T}(\hat{t}+)$. Also, since SG always executes Case S1.1.1 at a scheduling event which happens within $(\hat{t}, t)$ by the definition of $\hat{t}$, SG executes Case S1.1.1 $\mathcal{K}_1(\hat{t}+) - \mathcal{K}_1(t)$ times within $(\hat{t}, t)$. Therefore, $H^{(a)}_{OPT,1}(t) \geq H^{(a)}_{OPT,1}(\hat{t}+) - (\mathcal{K}_1(\hat{t}+) - \mathcal{K}_1(t)) = \mathcal{T}(\hat{t}+) - \mathcal{K}_1(\hat{t}+) + \mathcal{K}_1(t) = \mathcal{T}(\hat{t}+) - \mathcal{T}(\hat{t}+) + \mathcal{K}_1(t) > 0$.

Case 2. $\mathcal{K}_1(t) = 0$. First, we consider the case $\exists\tilde{t} \in (t', t]\ \mathcal{K}_1(\tilde{t}) > 0$. Let $\hat{t} \in (t', t]$ be the time when SG executes Case A2.2.1 and does not execute Case A2.2.1 within $(\hat{t}, t)$. By the definition of SG, $\mathcal{K}_1(\hat{t}+) = \mathcal{T}(\hat{t}+)$. By Lemma 3.2 and the above property (1), $H^{(a)}_{OPT,1}(\hat{t}+) = H^{(a)}_{SG,1}(\hat{t}+) = \mathcal{T}(\hat{t}+)$ since $h^{(a)}_{OPT}(\hat{t}+) = h^{(a)}_{SG}(\hat{t}+) = B$. Then, let t'' be the time when a scheduling event occurs such that $\forall t''' \in (t'', t]\ \ \mathcal{K}_1(t''-) > 0$ and $\mathcal{K}_1(t''') = 0$. Since $\mathcal{K}_2(\hat{t}+) = \mathcal{K}_3(\hat{t}+) = 0$ by the definition of $\hat{t}$, SG executes Case A2.1 and Case A1.1.1 $\mathcal{K}_2(t''+)$ and $\mathcal{K}_3(t''+)$ times, respectively, within $(\hat{t}, t'')$. The number of packets which SG (OPT) transmits from $Q^{(a)}$ within $(\hat{t}, t'')$ is at most $\mathcal{K}_1(\hat{t}+)$. So, $H^{(a)}_{OPT,1}(t''+) \geq H^{(a)}_{OPT,1}(\hat{t}+) + \mathcal{K}_2(t''+) + \mathcal{K}_3(t''+) - \mathcal{K}_1(\hat{t}+) \geq \mathcal{T}(\hat{t}+) + \mathcal{K}_2(t''+) + \mathcal{K}_3(t''+) - \mathcal{T}(\hat{t}+) \geq \mathcal{K}_2(t''+) + \mathcal{K}_3(t''+)$. Suppose that SG executes Case A2.1 and Case A1.1.1 within (t'', t) x and y times, respectively. By the above property (2) and (3), $x + y$ packets are accepted into S_1 of $Q^{(a)}$ by OPT within (t'', t). Then, the number of scheduling events which occur within (t'', t) is $\mathcal{K}_2(t''+) + x + \mathcal{K}_3(t''+) + y - \mathcal{K}_2(t) - \mathcal{K}_3(t)$. Therefore, $H^{(a)}_{OPT,1}(t) \geq H^{(a)}_{OPT,1}(t''+) + x + y - (\mathcal{K}_2(t''+) + x + \mathcal{K}_3(t''+) + y - \mathcal{K}_2(t) - \mathcal{K}_3(t)) \geq \mathcal{K}_2(t) + \mathcal{K}_3(t) > 0$.

Next, we consider the case $\forall\tilde{t} \in (t', t)\ \mathcal{K}_1(\tilde{t}) = 0$. Suppose that SG executes Case A2.1 and Case A1.1.1 z and w times, respectively, within (t', t). By the properties (2) and (3), $z+w$ packets are accepted into S_1 of $Q^{(a)}$ by OPT within (t', t). Then, since the number of scheduling events which happen within (t', t) is $\mathcal{K}_2(t'+)+z+\mathcal{K}_3(t'+)+w-\mathcal{K}_2(t)-\mathcal{K}_3(t)$, $H^{(a)}_{OPT,1}(t) \geq H^{(a)}_{OPT,1}(t'+) + \mathcal{K}_2(t'+) + z + \mathcal{K}_3(t'+) + w - (\mathcal{K}_2(t'+)+z+\mathcal{K}_3(t'+)+w-\mathcal{K}_2(t)+\mathcal{K}_3(t)) \geq \mathcal{K}_2(t)+\mathcal{K}_3(t) > 0$. □

Now, we show the following lemma using above lemmas.

Lemma 3.10: There exists an optimal offline algorithm which synchronizes with SG at any scheduling event when SG transmits a packet.

Proof. Consider an arbitrary optimal offline algorithm OPT. We consider scheduling events when SG transmits a packet from the head of the input σ, and if OPT does not synchronize with SG, then we modify OPT so that it synchronizes with SG.

Let t_0 be the time when a first scheduling event happens. Since we assume that OPT always transmits a packet at a scheduling event, it does so at t_0. Also, since there are no scheduling evetns before t_0, SG and OPT acts exactly the same way, the number of packets stored in each segment of both OTP and SG are exactly the same. Hence they can syncronize. So, SG can also transmit a packet at t_0. So, this is the first scheduling event when SG transmits a packet. As we impose the Synchronizing Rule to OPT, OPT synchronizes at t_0, namely the statement is true at t_0.

Let $t (\geq t_0)$ be a time when a scheduling event happens and SG transmits a packet, and assume that OPT synchronizes with SG within $(0, t-)$. We modify OPT so that it synchronize with SG at t.

Case 1: SG transmits a packet from S_1. First, we consider the case $\sum_{j=1}^{3} \mathcal{K}_j(t-) = 0$, namely, SG executes Case S1.2.2 at t. By the condition of Case S1.2.2, $H^{(1)}_{SG,0}(t-) = 0$ and $H^{(2)}_{SG,0}(t-) = 0$. Also, by Lemma 3.3, $H^{(1)}_{OPT,0}(t-) = H^{(1)}_{SG,0}(t-)$ and $H^{(2)}_{OPT,0}(t-) = H^{(2)}_{SG,0}(t-)$. So, OPT transmits a packet from S_1 at t. Hence, the statement is true. Next, we consider the case $\sum_{j=1}^{3} \mathcal{K}_j(t-) > 0$, namely, SG executes Case S1.1 at t. We assume OPT transmits a packet from $Q^{(a)}$ at t. If $H^{(a)}_{OPT,0}(t-) = 0$, OPT transmits a packet from S_1 since $H^{(a)}_{OPT,1}(t-) > 0$. Therefore, OPT and SG synchronize at t. On the other hand, If $H^{(a)}_{OPT,0}(t-) > 0$, $H^{(a)}_{OPT,1}(t-) > 0$ by Lemma 3.9. So, by the Synchronizing Rule, OPT transmits a packet from S_1 at t. Therefore, the statement is true.

Case 2: SG transmits a packet from S_0. We consider the case SG transmits a packet from S_0 of $Q^{(b)}$ at time t, namely, SG executes Case S1.2.1 at t. By the condition of Case S1.2.1, $\sum_{j=1}^{3} \mathcal{K}_j(t-) = 0$, $H^{(b)}_{SG,0}(t-) > 0$, and $H^{(a)}_{SG,0}(t-) = 0$ hold. First, assume that OPT transmits a packet from $Q^{(b)}$ at t. Since $H^{(b)}_{OPT,0}(t-) = H^{(b)}_{SG,0}(t-) > 0$ by Lemma 3.3, OPT transmits a packet from S_0 by the Synchronizing Rule. Hence, the statement is true. Hence, in what follows, we assume that OPT transmits a packet from $Q^{(a)}$ at t. Then, $H^{(a)}_{OPT,1}(t-) > 0$ holds and OPT transmits a packet from S_1 of $Q^{(a)}$ since $H^{(a)}_{OPT,0}(t-) = H^{(a)}_{SG,0}(t-) = 0$ by Lemma 3.3. Now, if OPT always transmits a packet from S_1 when it transmits from $Q^{(a)}$ within $(0, t)$, $H^{(a)}_{OPT,1}(t-) = 0$ holds by the fact $\sum_{j=1}^{3} \mathcal{K}_j(t-) = 0$ and Corollary 3.5, which contradicts the assumption that OPT trasmits a packet from S_1 of $Q^{(a)}$. Therefore, OPT transmits at least one packet from S_1 of $Q^{(b)}$ within $(0, t)$. Then, define $t' (< t)$ be a time when a scheduling event occurs such that OPT transmits a packet from S_1 of $Q^{(b)}$ and does not transmit a packet S_1 of $Q^{(b)}$ within (t', t). If a packet p which arrives to $Q^{(a)}$ at $\hat{t} \in (t', t)$ is accepted to S_1, p is transmitted within $(\hat{t}, t)$ by Lemma 3.4. Therefore, $\forall\tilde{t} \in [t'-, t)\ H^{(a)}_{OPT,1}(\tilde{t}) > 0$. If $\exists\tilde{t} \in (t', t)$, $h^{(b)}_{OPT}(\tilde{t}) = B$, $H^{(a)}_{OPT,1}(t-) = 0\ (a \neq b)$ by Lemma 3.7. This equality contradicts the above assumption that OPT transmits a packet from S_1 of $Q^{(a)}$. Hence, $\forall t'' \in (t', t)\ h^{(b)}_{OPT}(t'') < B$. Now, by the above three inequalities ($H^{(a)}_{OPT,1}(t'-) > 0$, $\forall t'' \in (t', t)\ h^{(b)}_{OPT}(t'') < B$, and $\forall\tilde{t} \in [t'-, t)\ H^{(a)}_{OPT,1}(\tilde{t}) > 0$), and Lemma 3.8, we may modify OPT into OPT' such that OPT' transmits a packet at t from S_1 of $Q^{(b)}$, transmits a packet at t' from S_1 of $Q^{(a)}$, and acts in the same way as OPT at any time other than t and t'. Since $H^{(b)}_{OPT',0}(t-) = H^{(b)}_{SG,0}(t-) > 0$ by Lemma 3.3, OPT'

can transmit a packet from S_0 of $Q^{(b)}$. Therefore, OPT' and SG synchronize at t, and the statement is true, which completes the proof. □

In the following analysis, we denote the optimal offline algorithm obtained by Lemma 3.10 as OPT^*. Now, we are ready to prove the main lemma.

Lemma 3.11: $F_{OPT^*,0}(\sigma) = F_{SG,0}(\sigma)$.

Proof. Recall that OPT^* and SG transmit a packet at a scheduling event whenever the buffer is not empty. So, using Lemma 3.3 and Lemma 3.10, we can conclude that OPT^* transmits a packet from S_0 at t if and only if SG transmits a packet from S_0 at t. □

3.3 Evaluating S_1

The analysis in this section goes as follows. Let σ be an input we are considering, and OPT^* be the optimal offline algorithm obtained in Sec. 3.2. We first construct another input σ' from σ. We then regard this σ' as an input for another problem, and prove that its optimal value is equal to $F_{OPT^*,1}(\sigma)$, which we want to estimate. Finally, we bound this optimal value.

Now, let us first explain how to construct σ' from σ. We use the following procedure.

(i) From σ, remove all scheduling events at which OPT^* does not transmit a packet from S_1. Let σ_1 be the resulting input.

(ii) From σ_1, remove all arrival events at which a packet that will be transmitted from S_0 arrives. Let σ_2 be the resulting input.

(iii) Let T be the time after the final event of σ happens. Then, since τ is 0 at the beginning, and it is incremented only when SG executes A1.1.1, SG executes A1.1.1 $\mathcal{T}(T)$ times for σ. Let t_k ($1 \leq k \leq \mathcal{T}(T)$) be the time when SG executes Case A1.1.1 kth time, and let p_k be the packet arrived at t_k. Also, let q_k be the mate of p_k, and r_k be the time when q_k arrived. (See the description of Case A1.1.1 in the algorithm in Sec. 2.2 for the definition of mate.) From σ_2, remove all $2\mathcal{T}(T)$ arrival events at t_k and r_k. Let σ_3 be the resulting input.

(iv) Before the first event of σ_3, add $\mathcal{T}(T)$ arrival events where arriving packets are destined for $Q^{(1)}$, and the same number of arrival events where arriving packets are destined for $Q^{(2)}$. Call these $2\mathcal{T}(T)$ packets $\mathcal{I}$-packets. Let σ' be the resulting input.

We then consider the 2-port Multi-queue buffer management problem on σ', where each buffer size is $\mathcal{T}(T)$ (hereafter, we call this model "a new model"). For an algorithm A for this problem, write its cost as $T_A(\sigma')$. Let OPT_τ be an optimal offline algorithm for this problem.

Lemma 3.12: $F_{OPT^*,1}(\sigma) = T_{OPT_\tau}(\sigma')$.

Proof. Note that σ_2 contains events for which OPT^* uses only S_1 when computing on σ, and the size of S_1 is at most $\mathcal{T}(T)$. In the new model, an algorithm has buffers each of which has size $\mathcal{T}(T)$. Hence, we can define the algorithm that acts, for each event on σ_2, exactly the same way as OPT^* for the corresponding event on σ. Let us call this algorithm A. From the above argument, it is easy to see that the cost of A on σ_2 is the same to that of OPT^* on σ, namely, $F_{OPT^*,1}(\sigma) = T_A(\sigma_2)$.

In the following, we extend A and define A' as follows: A' accepts all $\mathcal{I}$-packets at the beginning, and acts in the same way as A for other events. We show that $T_A(\sigma_2) = T_{A'}(\sigma')$, which completes the proof of the lemma since if this is true, A' transmits all packets of σ' and hence is an optimal offline algorithm for σ'. Recall that when SG executes A1.1.1, it increments τ, which means that the newly arriving packet and its mate exists in the largest position of S_1 at this moment. Hence, $2\mathcal{T}(T)$ packets p_k and q_k ($k = 1, \ldots, \mathcal{T}(T)$) are stored in different cells. Also, by the above argument, these packets are the first packets in σ_2 that arrive at the corresponding cell. Now, note that when constructing σ' from σ_2, we remove all these packets. Instead, we add $2\mathcal{T}(T)$ $\mathcal{I}$-packets that fill the whole buffer. It is then obvious that A' can accept all packets of σ' and hence $T_A(\sigma_2) = T_{A'}(\sigma')$. □

Next, we give the definition of an algorithm Greedy Algorithm(GR).

Greedy Algorithm (*GR*)**:** At a scheduling event at time t, if $h^{(1)}_{GR}(t-) \geq h^{(2)}_{GR}(t-)$, transmit a packet from $Q^{(1)}$. Otherwise, namely, if $h^{(1)}_{GR}(t-) < h^{(2)}_{GR}(t-)$, transmit a packet from $Q^{(2)}$.

In what follows, we show the relation between the number of packets which SG transmits from S_1 for σ and the number of packets which GR transmits for σ' in the new model.

Lemma 3.13: $F_{SG,1}(\sigma) = T_{GR}(\sigma')$.

Proof. Let $F_{SG,1}(\sigma, t)$ be the number of packets transmitted by SG for an input σ until t (including t), namely, $F_{SG,1}(\sigma, T) = F_{SG,1}(\sigma)$. Also, Let $T_{GR}(\sigma', t)$ be the number of packets transmitted by an algorithm GR for an input σ' until t (including t), namely, $T_{GR}(\sigma', T) = T_{GR}(\sigma')$. Also, let $e^{(i)}(t)$ (respectively, $E^{(i)}(t)$) be the number of empty cells in $Q^{(i)}$ of GR (respectively, S_1 of $Q^{(i)}$ of SG) at t, namely $e^{(i)}(t) = \mathcal{T}(T) - h^{(i)}_{GR}(t)$ (respectively, $E^{(i)}(t) = \mathcal{T}(t) - H^{(i)}_{SG,1}(t)$). Let t' be the time when the first event happens after the final $\mathcal{I}$-packet arrived in σ'. Note that the event at t' is also the first event of σ_2.

Now, we prove that the following equation holds, which gives the relation of the performances between SG for σ_2 and GR for σ': $\forall \tilde{t} \in [t', T]\ T_{GR}(\sigma', \tilde{t}) = F_{SG,1}(\sigma, \tilde{t})$. From this equation, we can obtain $T_{GR}(\sigma', T) = F_{SG,1}(\sigma, T)$, namely, $T_{GR}(\sigma') = F_{SG,1}(\sigma)$, which is exactly what we want to show. To make the induction proof simpler, we simultaneously prove the following: $\forall \tilde{t} \in [t', T] \forall i\ e^{(i)}(\tilde{t}) = E^{(i)}(\tilde{t})$.

Now, we start the proof of induction. We first show that the equalities are true at time $t'+$. A scheduling event does not happen by the construction method (i) of σ since OPT^* does not hold a packet at time $t'-$. Hence, $T_{GR}(\sigma', t'+) = T_{SG}(\sigma_2, t'+) = 0$, and An arrival event occurs at t. Recall that, by the construction of σ_2, $t' = r_1$ and q_1 is the arriving packet. Then, since SG accepts q_1 into S_0, $E^{(1)}(t'+) = 0$, and $E^{(2)}(t'+) = 0$. On the other hand, the event at t' is removed in σ' by the construction method (ii) of σ'. Hence, GR does nothing and so $T_{GR}(\sigma', t'+) = 0$, $e^{(1)}(t'+) = \mathcal{T}(T) - \mathcal{T}(T) + \mathcal{T}(T) - \mathcal{T}(T) = 0$, and $e^{(2)}(t'+) = \mathcal{T}(T) - \mathcal{T}(T) + \mathcal{T}(T) - \mathcal{T}(T) = 0$. By the above equalities, we have that $T_{GR}(\sigma', t'+) = F_{SG,1}(\sigma, t'+)$, $E^{(1)}(t'+) = e^{(1)}(t'+)$, and $E^{(2)}(t'+) = e^{(2)}(t'+)$.

Next, let $t(> t')$ be a time when an event happens in σ_2 (note that there is no time between t' and T where an event occurs only in σ'). We assume that the statement is true at time $t-$ and show that it is true at $t+$, namely, we assume that $T_{GR}(\sigma', t-) = T_{SG}(\sigma_2, t-)$, $E^{(1)}(t-) = e^{(1)}(t-)$, and $E^{(2)}(t-) = e^{(2)}(t-)$, and show that $T_{GR}(\sigma', t+) = T_{SG}(\sigma_2, t+)$, $E^{(1)}(t-) = e^{(1)}(t+)$, and $E^{(2)}(t-) = e^{(2)}(t+)$.

Case 1. Scheduling event (S1.1). By Lemma Appendix B.2, SG necessarily transmits a packet from S_1 since $\sum_{j=1}^{2} H_{SG,1}^{(j)}(t-) > 0$. $e^{(2)}(t-) - e^{(1)}(t-) = E^{(2)}(t-) - E^{(1)}(t-)$ holds since $\forall i\ E^{(i)}(t-) = e^{(i)}(t-)$ by the induction hypothesis. By the above equality and the definition of $E^{(i)}(t-)$ and $e^{(i)}(t-)$, $e^{(2)}(t-) - e^{(1)}(t-) = h_{GR}^{(1)}(t-) - h_{GR}^{(2)}(t-)$ and $E^{(2)}(t-) - E^{(1)}(t-) = H_{SG,1}^{(1)}(t-) - H_{SG,1}^{(2)}(t-)$. Then, we have that $h_{GR}^{(1)}(t-) - h_{GR}^{(2)}(t-) = H_{SG,1}^{(1)}(t-) - H_{SG,1}^{(2)}(t-)$ by the two above equalities. Next, at first, we consider the case where $h_{GR}^{(1)}(t-) - h_{GR}^{(2)}(t-) = H_{SG,1}^{(1)}(t-) - H_{SG,1}^{(2)}(t-) \geq 0$. Then, $E^{(1)}(t+) = E^{(1)}(t-) + 1$ and $E^{(2)}(t+) = E^{(2)}(t-)$ since SG transmits a packet from $Q^{(1)}$ by the definition of Greedy Step of SG. Also, $E^{(1)}(t-) < \mathcal{T}(t-) \leq \mathcal{T}(T)$ since SG has a packet at $Q^{(1)}$ at $t-$. By the induction hypothesis, $E^{(1)}(t-) = e^{(1)}(t-)$ holds. By the inequality and equality and the definition of $e^{(1)}(t-)$, $h_{GR}^{(1)}(t-) > 0$ holds, namely, GR has a packet at $Q^{(1)}$ at $t-$. Hence, by $h_{GR}^{(1)}(t-) - h_{GR}^{(2)}(t-) > 0$ and the definition of GR, GR transmits a packet form $Q^{(1)}$, namely, $e^{(1)}(t+) = e^{(1)}(t-) + 1$, and $e^{(2)}(t+) = e^{(2)}(t-)$. Therefore, by the above equality and the induction hypothesis, $E^{(1)}(t+) = e^{(1)}(t+) = E^{(1)}(t-) + 1 = e^{(1)}(t-) + 1$, and $E^{(2)}(t+) = e^{(2)}(t+) = E^{(2)}(t-) = e^{(2)}(t-)$. Also, by $T_{GR}(\sigma', t+) = T_{GR}(\sigma', t-) + 1$, $F_{SG,1}(\sigma, t+) = F_{SG,1}(\sigma, t-) + 1$, and the induction hypothesis, $T_{GR}(\sigma', t+) = F_{SG,1}(\sigma, t+)$.

Next, we consider the case where $h_{GR}^{(1)}(t-) - h_{GR}^{(2)}(t-) = H_{SG,1}^{(1)}(t-) - H_{SG,1}^{(2)}(t-) < 0$. By the definition of Greedy Step in SG, $E^{(1)}(t+) = E^{(1)}(t-)$, and $E^{(2)}(t+) = E^{(2)}(t-) + 1$ since SG transmits a packet from $Q^{(2)}$. By $h_{GR}^{(1)}(t-) - h_{GR}^{(2)}(t-) < 0$, GR has a packet at $Q^{(2)}$ at $t-$. Hence, by the definition of GR, GR also transmits a packet from $Q^{(2)}$, namely, $e^{(1)}(t+) = e^{(1)}(t-)$, and $e^{(2)}(t+) = e^{(2)}(t-) + 1$. Therefore, by the above equality and the induction hypothesis, $E^{(1)}(t+) = e^{(1)}(t+) = E^{(1)}(t-) + 1 = e^{(1)}(t-) + 1$, and $E^{(2)}(t+) = e^{(2)}(t+) = E^{(2)}(t-) = e^{(2)}(t-)$. Also, $T_{GR}(\sigma', t+) = F_{SG,1}(\sigma, t+)$ by the indcution hypothsis since $T_{GR}(\sigma', t+) = T_{GR}(\sigma', t-) + 1$, and $F_{SG,1}(\sigma, t+) = F_{SG,1}(\sigma, t-) + 1$.

Case 2. Scheduling event (S1.2.1). By the definition of SG, $E^{(1)}(t+) = E^{(1)}(t-)$, $E^{(2)}(t+) = E^{(2)}(t-)$, and $F_{SG,1}(\sigma, t+) = F_{SG,1}(\sigma, t-)$ since SG transmits a packet from S_0 at t. On the other hand, OPT^* transmits a packet from S_0 since OPT^* and SG synchronize at t. Hence, $e^{(1)}(t+) = e^{(1)}(t-)$, $e^{(2)}(t+) = e^{(2)}(t-)$, and $T_{GR}(\sigma', t+) = T_{GR}(\sigma', t-)$ since an event does not occur at t in σ' by the construction method (i) of σ'. By the above equalities and the induction hypothesis, $T_{GR}(\sigma', t+) = F_{SG,1}(\sigma, t+)$ and $\forall i\ e^{(i)}(t+) = E^{(i)}(t+)$.

Case 3. Scheduling event (S1.2.2). When SG has a packet in its buffer at $t-$, we can do the same argument as Case 1, and hence the proof is omitted.

Hence, we consider the case where there does not exist a packet in SG's buffer at $t-$, namely, $\forall i\ E^{(i)}(t-) = \mathcal{T}(t-)$ holds. When the buffer of SG is empty at $t-$, an arrival event does not happen after $t-$ by the assumption given in Sec. 2.1. Hence, $\mathcal{T}(t-) = \mathcal{T}(T)$ holds since SG does not execute Case A1.1.1 after $t-$. Therefore, $\forall i\ E^{(i)}(t-) = \mathcal{T}(T)$. Now, $\forall i\ e^{(i)}(t-) = \mathcal{T}(T)$ since $\forall i\ E^{(i)}(t-) = e^{(i)}(t-)$ by the induction hypothesis. Since $\forall i\ \mathcal{T}(T) - h_{GR}^{(i)}(t) = \mathcal{T}(T)$ by the definition of $e^{(i)}(t-)$, $h_{GR}^{(i)}(t) = 0$. Hence, the buffer of SG is also empty at $t-$. By the above argument, SG and GR do not transmit a packet, and the number of packets in each queue does not change. Therefore, $T_{GR}(\sigma', t+) = F_{SG,1}(\sigma, t+)$, and $\forall i\ E^{(i)}(t+) = e^{(i)}(t+)$.

Next, we consider cases where an arrival event happens at t. At an arrival event, since $T_{GR}(\sigma', t+) = T_{GR}(\sigma', t-)$ and $F_{SG,1}(\sigma, t+) = F_{SG,1}(\sigma, t-)$, we have that $T_{GR}(\sigma', t+) = F_{SG,1}(\sigma, t+)$ by the induction hypothesis.

Case 4. Arrival event (A1.1.1). By the definition of SG, since $\forall i\ H_{SG,1}^{(i)}(t+) = H_{SG,1}^{(i)}(t-) + 1$ and $\mathcal{T}(t+) = \mathcal{T}(t-) + 1$, $\forall i\ E^{(i)}(t+) = \mathcal{T}(t+) - H_{SG,1}^{(i)}(t+) = \mathcal{T}(t-) - H_{SG,1}^{(i)}(t-) = E^{(i)}(t-)$. On the other hand, $\forall i\ e^{(i)}(t+) = e^{(i)}(t-)$ since an event does not occur at t in σ' by the construction method (iii) of σ'. Therefore, $\forall i\ e^{(i)}(t+) = E^{(i)}(t+)$ by the induction hypothesis.

Case 5. Arrival event (A1.1.2). By the definition of SG, SG accepts an arriving packet p into S_0. Hence, $\forall i\ E^{(i)}(t+) = E^{(i)}(t-)$. On the other hand, an event does not occur at t in σ' by the construction method (ii) of σ' if p will be transmitted from S_0 by SG. Also, an event does not happen at t in σ' by the construction method (iii) of σ' if p will be transmitted from S_1 by SG. Therefore, $\forall i\ e^{(i)}(t+) = E^{(i)}(t+)$ by the induction hypothesis.

Case 6. Arrival event (A2.1). We assume that a packet p arrives at $Q^{(a)}$ at t. Then, by the definition of SG, $E^{(a)}(t+) = E^{(a)}(t-) - 1$ and $E^{(b)}(t+) = E^{(b)}(t-)$. Also, since $\forall i\ e^{(i)}(t-) = E^{(i)}(t-)$ by the induction hypothesis, GR can accept p when SG accepts p. Therefore, $e^{(a)}(t+) = e^{(a)}(t-) - 1$ and $e^{(b)}(t+) = e^{(b)}(t-)$ hold. Hence, $\forall i\ e^{(i)}(t+) = E^{(i)}(t+)$ by the induction hypothesis.

Case 7. Arrival event (A2.2.1). By the definition of SG, $\forall i\ E^{(i)}(t+) = E^{(i)}(t-)$ holds. Since $\forall i\ e^{(i)}(t-) = E^{(i)}(t-)$ by the induction hypothesis, GR rejects an arriving packet at t when SG rejects it. Hence, $\forall i\ e^{(i)}(t+) = e^{(i)}(t-)$. Therefore, we have that $\forall i\ e^{(i)}(t+) = E^{(i)}(t+)$ by the induction hypothesis.

We have shown that the statement is true at time $t+$. $\square$

Now, we analyze the competitive ratio of GR for σ' in the new model. Without loss of generality, we can assume that OPT accepts all arriving packets (General discussion can be found in p.302 of [11]). In what follows, we consider the case for large enough B and $\mathcal{T}(T)$. (In other cases, the competitive ratio of SG is smaller than $16/13$ in general. See Appendix C.)

Lemma 3.14: $T_{OPT_\tau}(\sigma') \le \frac{16}{13} T_{GR}(\sigma')$.

Proof. Let t' be the time for the first scheduling event. Note that $h^{(2)}_{GR}(t'-) = h^{(1)}_{GR}(t'-) = h^{(2)}_{OPT_\tau}(t'-) = h^{(1)}_{OPT_\tau}(t'-) = \mathcal{T}(T)$ by the construction of σ'. First of all, note that if GR does not reject a packet, GR gains the same cost as OPT_τ, in which case, the competitive ratio is 1.

So, suppose that GR rejects at least once, and let t_1 be the time when GR rejects an arriving packet for the first time. Let p be the rejected packet. For simplicity, we assume that p arrives at $Q^{(1)}$. (The other case, namely, the case that p arrives at $Q^{(2)}$ can be argued similarly, and hence we omit it here.) This means that $Q^{(1)}$ of GR is full at t_1-, namely, $h^{(1)}_{GR}(t_1-) = \mathcal{T}(T)$ (recall that buffer size is $\mathcal{T}(T)$). Hence $h^{(1)}_{GR}(t_1-) - h^{(1)}_{OPT_\tau}(t_1-) = \mathcal{T}(T) - h^{(1)}_{OPT_\tau}(t_1-)$. Let x_1 be this quantity. We define the time $\tilde{t}_1 (< t_1)$ for the scheduling event as follows: $h^{(1)}_{GR}(\tilde{t}_1-) - h^{(1)}_{OPT_\tau}(\tilde{t}_1-) = x_1 - 1$, $h^{(1)}_{GR}(\tilde{t}_1+) - h^{(1)}_{OPT_\tau}(\tilde{t}_1+) = x_1$. Since there is no packet only GR accepts and OPT and SG hold the same number of packets before t_1, $h^{(2)}_{OPT_\tau}(\tilde{t}_1+) - h^{(2)}_{GR}(\tilde{t}_1+) = h^{(1)}_{GR}(\tilde{t}_1+) - h^{(1)}_{OPT_\tau}(\tilde{t}_1+)$. By the definition of $\tilde{t}_1$, GR transmits a packet from $Q^{(2)}$ at $\tilde{t}_1$. Hence, by the definition of GR, $h^{(2)}_{GR}(\tilde{t}_1+) \ge h^{(1)}_{GR}(\tilde{t}_1+)$. From the above argument, $\mathcal{T}(T) \ge h^{(2)}_{OPT_\tau}(\tilde{t}_1+) = h^{(2)}_{GR}(\tilde{t}_1+) + x_1 \ge h^{(1)}_{GR}(\tilde{t}_1+) + x_1$ holds. So, GR transmits at least x_1 packets from $Q^{(1)}$, accepts at least x_1 packets before $\tilde{t}_1+$, and accepts at least x_1 packets within $(\tilde{t}_1, t_1)$. Since $\mathcal{T}(T) \ge h^{(2)}_{OPT_\tau}(\tilde{t}_1+) = h^{(2)}_{GR}(\tilde{t}_1+) + x_1 \ge h^{(1)}_{GR}(\tilde{t}_1+) + x_1 = h^{(1)}_{OPT_\tau}(\tilde{t}_1+) + 2x_1$, $\mathcal{T}(T) \ge 2x_1$ holds. Let c_1 be the number of packets that arrived within (t', t_1). From the above argument, $c_1 \ge x_1$ holds. GR accepts all these c_1 packets by the definition of t_1. Now, let $t''_1 (> t_1)$ be the time when an event happens such that $h^{(1)}_{GR}(t''_1-) = h^{(1)}_{OPT_\tau}(t''_1-) = \mathcal{T}(T)$ holds for the first time. The number of packets which OPT_τ accepts before t''_1- is $2\mathcal{T}(T) + c_1 + x_1$, and the number of packets which GR accepts before t''_1- is $2\mathcal{T}(T) + c_1$. If GR never rejects a packet after t''_1, $\frac{T_{OPT_\tau}(\sigma')}{T_{GR}(\sigma')} = \frac{2\mathcal{T}(T)+c_1+x_1}{2\mathcal{T}(T)+c_1} \le \frac{2\mathcal{T}(T)+2x_1}{2\mathcal{T}(T)+x_1} \le \frac{6}{5}$.

If GR rejects a packet after t''_1, let t_2 be the time when GR first rejects a packet after t''_1. We first consider the case that a packet arrives at $Q^{(1)}$ at t_2. Define x_2 as follows: At t_2-, $x_2 = h^{(2)}_{GR}(t_2-) - h^{(2)}_{OPT_\tau}(t_2-) = \mathcal{T}(T) - h^{(2)}_{OPT_\tau}(t_2-)$. We then define the scheduling time $\tilde{t}_2$ that satisfies $t''_1 < \tilde{t}_2 < t_2$, $h^{(1)}_{GR}(\tilde{t}_2-) - h^{(2)}_{OPT_\tau}(\tilde{t}_2-) = x_2 - 1$, and $h^{(1)}_{GR}(\tilde{t}_2+) - h^{(2)}_{OPT_\tau}(\tilde{t}_2+) = x_2$. Since there are no packets only GR accepts, $h^{(2)}_{OPT_\tau}(\tilde{t}_2+) - h^{(2)}_{GR}(\tilde{t}_2+) = x_1 + h^{(1)}_{GR}(\tilde{t}_2+) - h^{(1)}_{OPT_\tau}(\tilde{t}_2+)$. By the definition of $\tilde{t}_2$, GR transmits a packet from $Q^{(2)}$ at $\tilde{t}_2$. Hence by the definition of GR, $h^{(2)}_{GR}(\tilde{t}_2+) \ge h^{(1)}_{GR}(\tilde{t}_2+)$ holds. From the above arguments, $\mathcal{T}(T) \ge h^{(2)}_{OPT_\tau}(\tilde{t}_2+) = h^{(2)}_{GR}(\tilde{t}_2+) + x_1 + x_2 \ge h^{(1)}_{GR}(\tilde{t}_1+) + x_1 + x_2$ holds. So, GR transmits at least $x_1 + x_2$ packets from $Q^{(1)}$ within $(t''_1, \tilde{t}_2+)$, and accepts at least $x_1 + x_2$ packets into $Q^{(1)}$ within $(\tilde{t}_2, t_2)$. Futhermore, since $\mathcal{T}(T) \ge h^{(2)}_{OPT_\tau}(\tilde{t}_2+) = h^{(2)}_{GR}(\tilde{t}_2+) + x_1 + x_2 \ge h^{(1)}_{GR}(\tilde{t}_1+) + x_1 + x_2 = h^{(1)}_{OPT_\tau}(\tilde{t}_1+) + x_1 + 2x_2$, $\mathcal{T}(T) \ge x_1 + 2x_2$ holds. Let c_2 be the number of packets that arrived within (t''_1, t_2). Then, from the above arguments, $c_2 \ge x_1 + x_2$ holds, and by the definition of t_2, GR accepts all these c_2 packets. Also, define $t''_2 (> t_2)$, as the first time $h^{(2)}_{GR}(t''_2-) = h^{(2)}_{OPT_\tau}(t''_2-) = \mathcal{T}(T)$ holds. The number of packets OPT_τ accepted before t''_2- is $2\mathcal{T}(T) + c_1 + x_1 + c_2 + x_2$, and the number of packets GR accepted before t''_2- is $2\mathcal{T}(T) + c_1 + c_2$. If GR accepts all arriving packets after t''_2, $\frac{T_{OPT_\tau}(\sigma')}{T_{GR}(\sigma')} = \frac{2\mathcal{T}(T)+c_1+x_1+c_2+x_2}{2\mathcal{T}(T)+c_1+c_2} \le \frac{2\mathcal{T}(T)+3x_1+2x_2}{2\mathcal{T}(T)+2x_1+x_2} \le \frac{3\mathcal{T}(T)+2x_1}{\frac{5}{2}\mathcal{T}(T)+\frac{3}{2}x_1} \le \frac{16}{13}$ holds. We can do a similar argument for the case that a packet arrives at $Q^{(1)}$ at t_1, and a packet arrives at $Q^{(2)}$ at t_2.

Finally, we do not have to consider the case that GR rejects a packet after t''_2 by Corollary 1 in Sec.4 in [11]. $\square$

Lemma 3.15: $F_{OPT^*,1}(\sigma) \le \frac{16}{13} F_{SG,1}(\sigma)$.

Proof. Using Lemma 3.12, Lemma 3.13 and Lemma 3.14, the statement is true. $\square$

4. Concluding Remarks

In this paper, we proposed the algorithm SG in the multi-queue switches model for $m = 2$, and proved that its competitive ratio is $\frac{16}{13}$. This matches the lower bound given by Schmidt [11].

References

[1] W. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosén, "Competitive queue policies for differentiated services," J. Algorithms, vol.55, no.2, pp.113–141, 2005.

[2] S. Albers and T. Jacobs, "An experimental study of new and known online packet buffering algorithms," Proc. 15th Annual European Symposium on Algorithms, pp.754–765, 2007.

[3] S. Albers and M. Schmidt, "On the performance of greedy algorithms in packet buffering," SIAM J. Comput., vol.35, no.2, pp.278–304, 2005.

[4] Y. Azar and A. Litichevskey, "Maximizing throughput in multi-queue switches," Algorithmica, vol.45, no.1, pp, 69–90, 2006.

[5] Y. Azar and Y. Richter, "Management of multi-queue switches in QoS networks," Algorithmica, vol.43, no.1-2, pp, 81–96, 2005.

[6] Y. Azar and Y. Richter, "The zero-one principle for switching networks," Proc. 36th Annual ACM Symposium on Theory of Computing, pp.64–71, 2004.

[7] M. Bienkowski and A. Madry, "Geometric aspects of online packet buffering: An optimal randomized algorithm for two buffers," Proc. 8th Latin American Theoretical INformatics, pp.252–263, 2008.

[8] A. Borodin and R. El-Yaniv, Online computation and competitive analysis, Cambridge University Press, 1998.

[9] T. Itoh and T. Nagumo, "Improved lower bounds for competitive ratio of multi-queue switches in QoS networks," IEICE Trans. Fundamentals, vol.E88-A, no.5, pp.1155–1165, May 2005.
[10] T. Itoh and N. Takahashi, "Competitive analysis of multi-queue preemptive QoS algorithms for general priorities," IEICE Trans. Fundamentals, vol.E89-A, no.5, pp.1186–1197, May 2006.
[11] M. Schmidt, "Packet buffering: Randomization beats deterministic algorithms," Proc. 22nd International Symposium on Theoretical Aspects of Computer Science, pp.293–304, 2005.
[12] D. Sleator and R. Tarjan, "Amortized efficiency of list update and paging rules," CACM 28, pp.202–208, 1985.

Appendix A: Example of an Execution of *SG*

Here, we explain behavior of SG using an example. We consider the case $B = 8$. The left top of Fig. A· 1 illustrates buffers. The upper and the lower rows correspond to $Q^{(1)}$ and $Q^{(2)}$, respectively. For each cell, the label (i, j) is given, which means that the cell is the jth position of $Q^{(i)}$, namely, $Q^{(i,j)}$. The bold vertical bar represents a border between S_0 and S_1.

The input is given in the top right table. In this example, each event happens at an integer timeslot. For example, at times 1, 2, and 3, arrival events happen, and arriving packets are destined for $Q^{(2)}$, $Q^{(2)}$, and $Q^{(1)}$, respectively. Then at time 4, a scheduling event happens. Corresponding to each event, the column denoted by "action" shows which case SG executes.

At times 1 and 2, packets a and b arrive at $Q^{(2)}$. SG executes Case A1.1.2 and accepts these packets. (In the figure, a packet just accepted is highlighted by a square.) Next, at time 3, the packet c arrives at $Q^{(1)}$, and SG accepts it. This time, Case A1.1.1 is executed since the first position of the other queue already holds a packet. Then, SG increments the value of τ, and the size of S_0 decreases by 1 (and accordingly, the S_1 part appears). Now, the current values of k_1, k_2, and k_3 are 0, 0, and 1, respectively.

At time 4, SG executes a transmission. Since $k_1 + k_2 + k_3 > 0$, it selects Case S1.1, and since $k_1 = k_2 = 0$ and $k_3 = 1$. it selects Case S1.1.3. So, SG executes Greedy Step. Since $H^{(1)}_{SG,1}(t-) = H^{(2)}_{SG,1}(t-) = 1$, it selects $Q^{(1)}$, and the pacekt c is transmitted. Each transmitted packet is denoted by a black square. In addition, k_3 is decremented by one.

In this way, the computation continues up to time 24. We stop detailed explanation because of space restriction.

Appendix B: The Properties of *SG*

In this section, we show two lemmas about the properties of SG.

Lemma Appendix B.1: At any time t, if $H^{(a)}_{SG,0}(t) > 0$, $H^{(b)}_{SG,0}(t) = 0$ $(a \neq b)$.

Proof. We show the proof by induction. At the beginning, S_0 of both queues are empty. First, note that at scheduling events, the invariant is not broken, so consider arrival events. When a packet is stored in S_0, if both S_0 are empty, then the invariant still holds. So, suppose that S_0 of one queue, say $Q^{(a)}$, is empty but S_0 of the other queue, say $Q^{(b)}$, is non-empty. If a new packet is stored in S_0 of $Q^{(b)}$, the invariant holds again since $Q^{(a)}$ is still empty. If a new packet is stored in S_0 of $Q^{(a)}$, then Case A1.1.1 is executed, and τ is incremented by one. As a result, the cell for the newly stored packet moves from S_0 to S_1 and hence S_0 of $Q^{(a)}$ becomes empty again. This completes the proof. □

Lemma Appendix B.2:
$\forall t\ \sum_{i=1}^{2} H^{(i)}_{SG,1}(t) \geq \sum_{j=1}^{3} \mathcal{K}_j(t)$.

Proof. We prove the lemma inductively on events. Let t_1 be the time when the first event happens. When a scheduling event occurs at t_1, Case S1.2 is executed at t_1 and $\sum_{j=1}^{3} \mathcal{K}_j(t_1+) = \sum_{j=1}^{3} \mathcal{K}_j(0) = 0$ holds. Also, since there does not exist a packet at t_1, $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t_1+) = \sum_{i=1}^{2} H^{(i)}_{SG,1}(0) = 0$. When an arrival event happens at t_1, Case A1.1.2is executed at t_1 and $\sum_{j=1}^{3} \mathcal{K}_j(t_1+) = \sum_{j=1}^{3} \mathcal{K}_j(0) = 0$. By the definition of SG, since the packet arriving at t_1 is accepted to S_0, $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t_1+) = \sum_{i=1}^{2} H^{(i)}_{SG,1}(0) = 0$. Therefore, the statement is true at t_1.

Let t be a time when an event happens. We assume that the statement is true at time $t-$ and show that it is true at $t+$, namely, we assume that $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t-) \geq \sum_{j=1}^{3} \mathcal{K}_j(t-)$ and show $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \sum_{j=1}^{3} \mathcal{K}_j(t+)$.

At first, we consider the case where a scheduling event happens at t.

Case 1. Scheduling event (S1.1). By the definition of SG, $\sum_{j=1}^{3} \mathcal{K}_j(t-) \geq 1$ and $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-) - 1$. Also, $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t-) \geq \sum_{j=1}^{3} \mathcal{K}_j(t-)$ by the induction hypothesis. Therefore, since $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \sum_{i=1}^{2} H^{(i)}_{SG,1}(t-) - 1$, $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \sum_{j=1}^{3} \mathcal{K}_j(t+)$ by the above equalities.

Case 2. Scheduling event (S1.2). By the definition of SG, since $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-) = 0$, the statement is true.

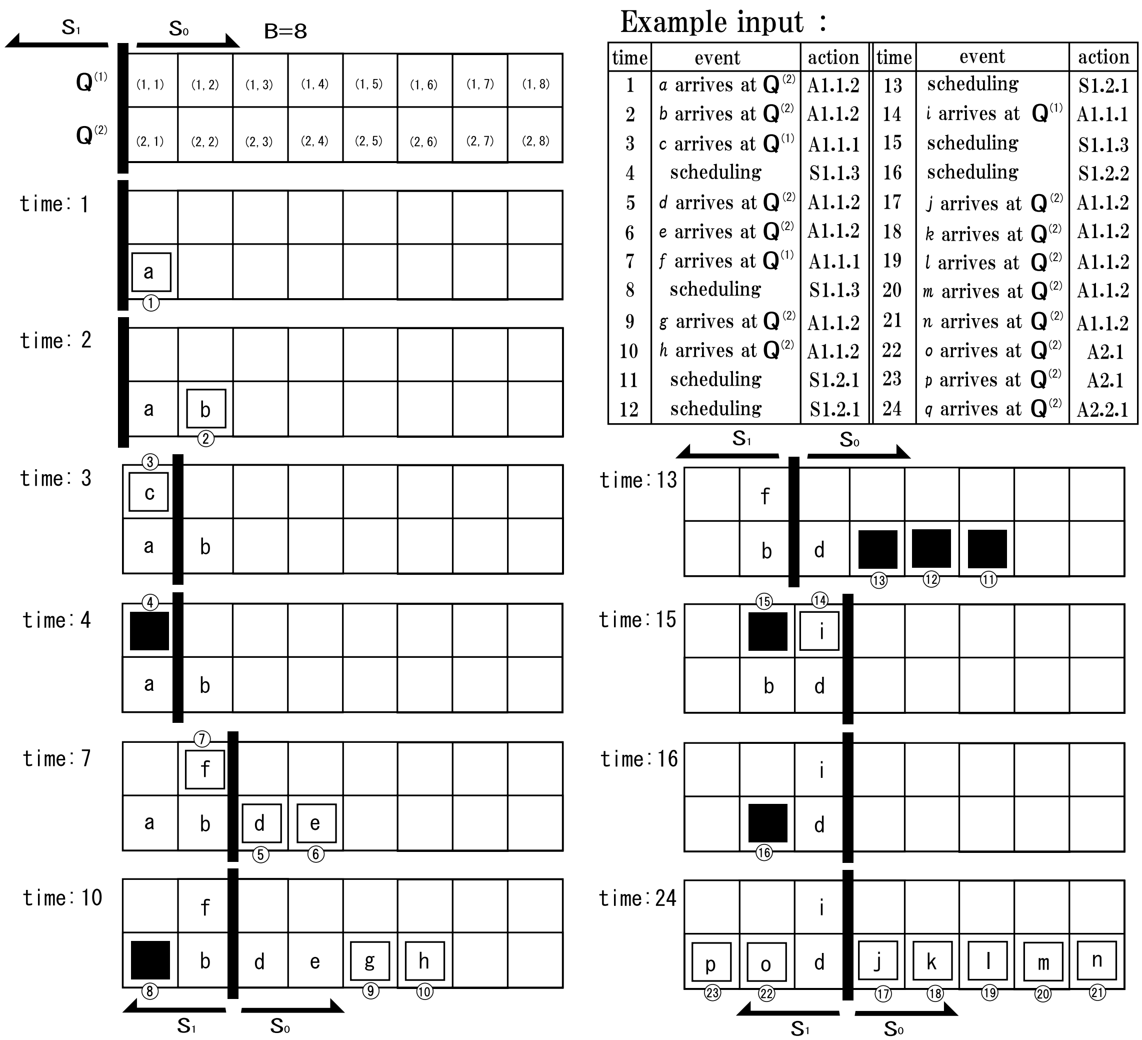
Next, we consider the case where an arrival event occurs at t.

Case 3. Arrival event (A1.1.1, A2.1). By the definition of SG, $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \sum_{i=1}^{2} H^{(i)}_{SG,1}(t-) + 2$ and $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-) + 1$. By the induction hypothesis,
$\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \sum_{j=1}^{3} \mathcal{K}_j(t+)$.

Case 4. Scheduling event (A1.1.2). By the definition of SG, $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) = \sum_{i=1}^{2} H^{(i)}_{SG,1}(t-)$ and $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \sum_{j=1}^{3} \mathcal{K}_j(t-)$. Therefore, by the induction hypothesis,
$\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \sum_{j=1}^{3} \mathcal{K}_j(t+)$.

Case 5. Scheduling event (A2.2.1). By the definition of SG, $\sum_{j=1}^{3} \mathcal{K}_j(t+) = \mathcal{T}(T)$ holds, and $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \mathcal{T}(T)$ since there exists at least one queue of SG which is full. Hence, $\sum_{i=1}^{2} H^{(i)}_{SG,1}(t+) \geq \sum_{j=1}^{3} \mathcal{K}_j(t+)$.

We have shown that the statement is true at time $t+$. □



time	event	action	time	event	action
1	*a* arrives at $Q^{(2)}$	A1.1.2	13	scheduling	S1.2.1
2	*b* arrives at $Q^{(2)}$	A1.1.2	14	*i* arrives at $Q^{(1)}$	A1.1.1
3	*c* arrives at $Q^{(1)}$	A1.1.1	15	scheduling	S1.1.3
4	scheduling	S1.1.3	16	scheduling	S1.2.2
5	*d* arrives at $Q^{(2)}$	A1.1.2	17	*j* arrives at $Q^{(2)}$	A1.1.2
6	*e* arrives at $Q^{(2)}$	A1.1.2	18	*k* arrives at $Q^{(2)}$	A1.1.2
7	*f* arrives at $Q^{(1)}$	A1.1.1	19	*l* arrives at $Q^{(2)}$	A1.1.2
8	scheduling	S1.1.3	20	*m* arrives at $Q^{(2)}$	A1.1.2
9	*g* arrives at $Q^{(2)}$	A1.1.2	21	*n* arrives at $Q^{(2)}$	A1.1.2
10	*h* arrives at $Q^{(2)}$	A1.1.2	22	*o* arrives at $Q^{(2)}$	A2.1
11	scheduling	S1.2.1	23	*p* arrives at $Q^{(2)}$	A2.1
12	scheduling	S1.2.1	24	*q* arrives at $Q^{(2)}$	A2.2.1

Fig. A·1 Example of an execution of SG.

Appendix C: Competitive Ratios for Small $\mathcal{T}(T)$

To analyze the competitive ratio $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} = \frac{F_{OPT,0}(\sigma)+F_{OPT,1}(\sigma)}{F_{SG,0}(\sigma)+F_{SG,1}(\sigma)}$, we estimated the gains incurred by S_0 and S_1 separately when we considered large enough B and $\mathcal{T}(T)$. However, in the case where $\mathcal{T}(T)$ is not large for B, the above analysis does not work, namely, the ratio we obtain becomes larger than the actual value. To overcome this, we will adopt a different way of analysis, as seen in the following corollary. Namely, we evaluate the gains of S_0 and S_1 together, as we have done in the proof of Lemma 3.14.

Corollary Appendix C.1: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{2B+2\mathcal{T}(T)}{2B+\frac{5}{4}\mathcal{T}(T)}$.

Proof(sketch). The proof is similar to that of Lemma 3.14: The number of packets accepted by OPT^* (SG, respectively) to S_1 before $t_2''-$ is estimated as $2\mathcal{T}(T)+c_1+x_1+c_2+x_2$ ($2\mathcal{T}(T)+c_1+c_2$, respectively). On the other hand, SG executes Case A2.2.1 before $t_1''-$ and within (t_1'', t_2''). Hence, OPT^* and SG accept $2(B-\mathcal{T}(T))$ packets to S_0 before $t_2''-$. Therefore, we have that $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} = \frac{F_{OPT,0}(\sigma)+F_{OPT,1}(\sigma)}{F_{SG,0}(\sigma)+F_{SG,1}(\sigma)} \le \frac{2(B-\mathcal{T}(T))+3\mathcal{T}(T)+2x_1}{2(B-\mathcal{T}(T))+\frac{1}{2}\mathcal{T}(T)+\frac{3}{2}x_1} \le \frac{2(B-\mathcal{T}(T))+2\mathcal{T}(T)+c_1+x_1+c_2+x_2}{2(B-\mathcal{T}(T))+2\mathcal{T}(T)+c_1+c_2} \le \frac{2B+2\mathcal{T}(T)}{2B+\frac{5}{4}\mathcal{T}(T)}$. □

By this corollary, in the case where B is larger than $\mathcal{T}(T)$, namely, in the case where B is large enough and $\mathcal{T}(T)$ is small, we have that $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \simeq 1$. In what follows, we can show the competitive ratio for small B and small $\mathcal{T}(T)$ in the same way as the proof of Lemma 3.14. Note that $B \ge \mathcal{T}(T)$ and $F_{OPT,0}(\sigma) = F_{SG,0}(\sigma) \ge B - \mathcal{T}(T)$.

$\mathcal{T}(T) = 0$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} = 1$.

$\mathcal{T}(T) = 1$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{B-1+3}{B-1+2} = \frac{B+2}{B+1} \le \frac{3}{2} = 1.5$.

$\mathcal{T}(T) = 2$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{B-2+8}{B-2+6} = \frac{B+6}{B+4} \le \frac{8}{6} \simeq 1.334$.

$\mathcal{T}(T) = 3$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{B-3+12}{B-3+9} = \frac{B+9}{B+6} \le \frac{12}{9} \simeq 1.334$.

$\mathcal{T}(T) = 4$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{B-4+20}{B-4+16} = \frac{B+16}{B+12} \le \frac{20}{16} = 1.25$.

$\mathcal{T}(T) = 5$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{B-5+25}{B-5+20} = \frac{B+20}{B+15} \le \frac{25}{20} = 1.25$.

$\mathcal{T}(T) = 6$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{B-6+24}{B-6+19} = \frac{B+18}{B+13} \le \frac{24}{19} \simeq 1.264$.

$\mathcal{T}(T) = 7$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \le \frac{B-7+28}{B-7+22} = \frac{B+21}{B+15} \le \frac{28}{22} = 1.273$.

$\mathcal{T}(T) = 8$: $\frac{T_{OPT^*}(\sigma)}{T_{SG}(\sigma)} \leq \frac{B-8+32}{B-8+26} = \frac{B+24}{B+18} \leq \frac{16}{13} \simeq 1.231.$

Koji Kobayashi is a doctor course student in the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University. He received B.E. and M.E. degrees from Kyoto University in 2004 and 2006, respectively. His research interest is online algorithms.

Shuichi Miyazaki is an associate professor of Academic Center for Computing and Media Studies, Kyoto University, Kyoto, Japan. He received BE, ME, and Ph.D. degrees from Kyushu University in 1993, 1995 and 1998, respectively. His research interests include algorithms and complexity theory.

Yasuo Okabe is a professor of Academic Center for Computing and Media Studies, Kyoto University. He received BE, ME and Ph.D. from Kyoto University in 1986, 1988, 1994 respectively. His current research interest includes Internet architecture, ubiquitous networking and distributed algorithms. He is a member of IPSJ, JSSST, IEEE, ACM and EATCS.